



Insper

Ativos Digitais e Blockchain

Ricardo Rocha
Raul Ikeda

Objetivo

- Programando SmartContracts

Relembrando

Site: <http://ethereum.org/developers/>

- Smart Contract Language:
 - Solidity (javascript)
 - Bamboo
 - Vyper (tipo Python)
 - Flint
- Compiler:
 - Truffle
 - Waffle
 - Brownie
 - etc
- Network:
 - Ganache
 - Ethnode
 - Infura
 - etc.

Teste rápido:

1. Acessar: <https://etherpad.opendev.org/p/AD-Test>
 1. Colocar o nome na frente de um endereço
 2. (14) 0x7340caCed7Dd7046f6a3946F934b8111753fB0a0 **Raul**
2. Acessar: <http://remix.ethereum.org/>
3. Fazer o upload do arquivo auction.sol do Blackboard
4. Abrir o arquivo no remix e compilar
5. Na Aba Deploy & Run Transactions:
 1. Selecionar o Environment: Web3 Provider
 2. Colocar o endpoint: <http://18.207.173.244:8545>
 3. Selecionar o account que você marcou no etherpad
 4. Preencher o campo Load Contract from Address:
 1. 0XfEFf653e81EA3282C3Efa664641865077ed5A2C7
 5. Apertar o Botão At Address
 6. Aguardar o início do leilão

Contrato Auction.sol

Atributos:

- beneficiary: endereço do vendedor
- auctionEndTime: tempo de término do leilão
- highestBidder: endereço do dono da maior oferta
- highestBid: valor da maior oferta
- pendingReturns: dicionário de ofertas

Métodos:

- constructor: roda automaticamente quando implanta o contrato
- bid(): registra uma oferta
- withdraw(): saca o dinheiro das ofertas, exceto o comprador
- auctionEnd(): encerra o leilão

Eventos:

- HighestBidIncreased: quando há um novo best bid
- AuctionEnded: quando o leilão acaba

Testando Rapidamente usando Remix

No próprio Remix:

- Alterar Environment para JavaScript VM
- Selecionar uma conta
- Fazer o Deploy preenchendo `_BIDDINGTIME` e `_BENEFICIARY`
- Testar as funções novamente

Montagem do Ambiente

- Instalar o Nodejs 12.16.3 LTS
 - <https://nodejs.org/en/>
 - Quem está no Ubuntu: atualizar usando npm
- Instalar o Truffle (No prompt):
 - `npm install truffle -g`
- Instalar o Ganache
 - github.com/trufflesuite/ganache/releases/download/v2.4.0/Ganache-2.4.0-win-setup.exe
- Instalar Metamask:
 - <https://metamask.io/>

Testando usando Ganache

- Crie uma pasta e acesse via prompt
- `> truffle init`
- Coloque o arquivo `auction.sol` em `contracts`
- `> truffle compile`
- Abrir o arquivo `truffle-config.js`:
 - Remover o comentário do bloco `development`
- `> truffle migrate`
- Um Frontend faz falta agora!

Tutorial

Realizar o tutorial:

- <https://www.trufflesuite.com/tutorials/pet-shop>

Cuidado para configurar o MetaMask corretamente

ATENÇÃO: Se você já possui o MetaMask e se você possui Ethers na rede principal, muito cuidado na hora de “gastar” a simulação. Veja bem qual rede e qual conta você está usando.

Extra: <https://www.trufflesuite.com/tutorials/building-testing-frontend-app-truffle-3>

Próxima Aula

Continuação do desenvolvimento de SmartContracts

Próxima aula

- Tutorial de Solidity